

ĐẠI HỌC QUỐC GIA TP HCM  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA ĐIỆN TỬ - VIỄN THÔNG

## Báo cáo tuần 6

### Bài 9: Thực hành tổng hợp (2 tuần)

Môn học: Thực hành Phương pháp tính

Sinh viên thực hiện:  
Đặng Xuân Nam - 22200106

Giáo viên hướng dẫn:  
Phạm Thế Hùng

Ngày 23 tháng 12 năm 2024



## Mục lục

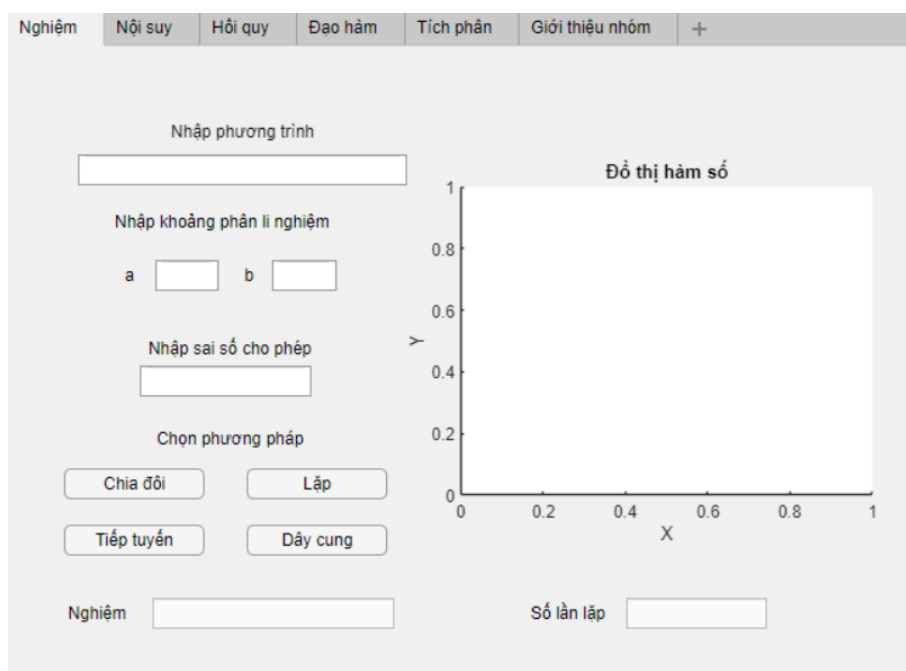
<b>1</b>	<b>Tính gần đúng hệ phương trình</b>	<b>3</b>
1.1	Giao diện	3
1.2	Thuật toán	3
1.3	Kiểm tra	8
<b>2</b>	<b>Nội suy</b>	<b>9</b>
2.1	Giao diện	9
2.2	Thuật toán	9
2.3	Kiểm tra	13
<b>3</b>	<b>Hồi quy</b>	<b>14</b>
3.1	Giao diện	14
3.2	Thuật toán	14
3.3	Kiểm tra	16
3.3.1	Tuyến tính	16
3.3.2	Hàm mũ	16
<b>4</b>	<b>Đạo hàm</b>	<b>17</b>
4.1	Giao diện	17
4.2	Thuật toán	17
4.3	Kiểm tra	22
<b>5</b>	<b>Tích phân</b>	<b>24</b>
5.1	Giao diện	24
5.2	Thuật toán	24
5.3	Kiểm tra	28
5.3.1	Phương pháp hình thang	28
5.3.2	Phương pháp Simpson's 1/3	28
5.3.3	Phương pháp Simpson's 3/8	29

## Danh sách hình vẽ

1	Giao diện Tìm nghiệm . . . . .	3
2	Thuật toán phương pháp chia đôi. . . . .	4
3	Thuật toán phương pháp lặp. . . . .	5
4	Thuật toán pháp tiếp tuyến. . . . .	6
5	Thuật toán phương pháp dây cung. . . . .	7
6	Sử dụng phương pháp chia đôi . . . . .	8
7	Sử dụng phương pháp lặp . . . . .	8
8	Sử dụng phương pháp tiếp tuyến . . . . .	8
9	Sử dụng phương pháp dây cung . . . . .	8
10	Giao diện Nội suy . . . . .	9
11	Sơ đồ giải thuật phương pháp Lagrange. . . . .	11
12	Kết quả bằng phương pháp Lagrange . . . . .	13
13	Kết quả bằng phương pháp Newton (lùi) . . . . .	13
14	Kết quả bằng phương pháp Newton (tiến) . . . . .	13
15	Giao diện Hồi quy . . . . .	14
16	Kết quả hồi quy bằng tuyến tính . . . . .	16
17	Kết quả hồi quy bằng hàm mũ . . . . .	17
18	Giao diện Đạo hàm . . . . .	17
19	Xấp xỉ lùi với sai số $O(h)$ . . . . .	22
20	Xấp xỉ lùi với sai số $O(h^2)$ . . . . .	22
21	Xấp xỉ trung tâm với sai số $O(h)$ . . . . .	22
22	Xấp xỉ trung tâm với sai số $O(h^2)$ . . . . .	22
23	Xấp xỉ tiến với sai số $O(h)$ . . . . .	22
24	Xấp xỉ tiến với sai số $O(h^2)$ . . . . .	22
25	Xấp xỉ lùi với sai số $O(h)$ . . . . .	23
26	Xấp xỉ lùi với sai số $O(h^2)$ . . . . .	23
27	Xấp xỉ trung tâm với sai số $O(h)$ . . . . .	23
28	Xấp xỉ trung tâm với sai số $O(h^2)$ . . . . .	23
29	Xấp xỉ tiến với sai số $O(h)$ . . . . .	23
30	Xấp xỉ tiến với sai số $O(h^2)$ . . . . .	23
31	Giao diện Tích phân . . . . .	24
32	Công thức tích phân hình thang dạng tổng quát . . . . .	26
33	Công thức tích phân Simpson 1/3. . . . .	26
34	Công thức tích phân Simpson 3/8. . . . .	27
35	Kết quả của nhập x và y . . . . .	28
36	Kết quả của nhập hàm $y = x^3 \sin(x) + x \cos(x)$ . . . . .	28
37	Kết quả của nhập x và y . . . . .	28
38	Kết quả của nhập hàm $y = x$ . . . . .	28
39	Kết quả của nhập x và y . . . . .	29
40	Kết quả của nhập hàm $y = x^3 \sin(x) + x \cos(x)$ . . . . .	29

# 1 Tính gần đúng hệ phương trình

## 1.1 Giao diện



Hình 1: Giao diện Tìm nghiệm

## 1.2 Thuật toán

```
1 % Value changed function: EditFieldPT
2 function EditFieldPTValueChanged(app, event)
3     global phươngtrinh;
4     phươngtrinh = app.EditFieldPT.Value;
5     phươngtrinh = str2func(['@(x)', phươngtrinh]);
6 end
```

Listing 1: Hàm nhận phương trình tìm nghiệm

```
1 % Value changed function: EditFielda
2 function EditFieldaValueChanged(app, event)
3     global a;
4     a = app.EditFielda.Value;
5     a = str2double(a);
6 end
```

Listing 2: Hàm nhận giá trị a

```
1 function EditFieldbValueChanged(app, event)
2     global b;
3     b = app.EditFieldb.Value;
4     b = str2double(b);
5 end
```

Listing 3: Hàm nhận giá trị b

```
1 % Value changed function: EditFielde
2 function EditFieldeValueChanged(app, event)
3     global e;
```

```

4     e = app.EditFielde.Value;
5     e = str2double(e);
6 end

```

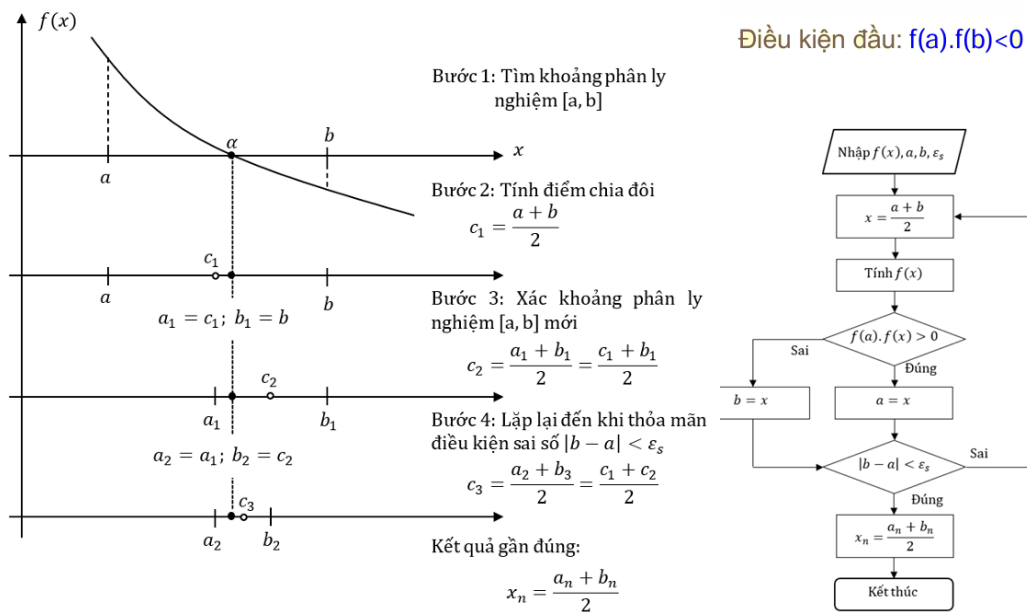
Listing 4: Hàm nhận giá trị sai số cho phép

```

1 % Button pushed function: ButtonChiaDoi
2 function ButtonChiaDoiPushed(app, event)
3     global a;
4     global phuongtrinh;
5     global e;
6     global b;
7     %-----
8     x = linspace(a,b,100);
9     y = phuongtrinh(x);
10    A = a; B = b;
11    solanlap = 0;
12    nghiem = 0;
13    while abs(B-A) >= e
14        nghiem = (A+B)/2;%chon nghiem nam giữa khoảng phân li
15        %xet nghiem tồn tại giữa a và nghiem
16        if phuongtrinh(nghiem)*phuongtrinh(A) > 0
17            A = nghiem;%vi nghiem không tồn tại trong [a;nghiem]
18        else
19            B = nghiem;%vi nghiem tồn tại trong [a;b]
20        end
21        solanlap = solanlap + 1;
22    end
23    %-----
24    figure('Visible','off');
25    plot(app.UIAxes,x,y,nghiem,'r*');
26    app.EditFieldNghiem.Value = sprintf('%.4f',nghiem);
27    app.EditFieldSoLanLap.Value = sprintf('%d',solanlap);
28 end

```

Listing 5: Hàm thực hiện phương pháp chia đôi



Hình 2: Thuật toán phương pháp chia đôi.

```

1 % Button pushed function: ButtonLap

```

```

2 function ButtonLapPushed(app, event)
3     global a;
4     global phuongtrinh;
5     global e;
6     global b;
7 %-----
8     x = linspace(a,b,100);
9     y = phuongtrinh(x);
10    A = a; B = b;
11    solanlap = 0;
12    nghiem = (A + B)/2; %chon nghiem nam giua [a;b]
13    while (true)
14        z = phuongtrinh(nghiem);% x_n = g(x_{n-1})
15        if abs(z - nghiem) < e %|x_n - g(x_{n-1})| < saiso
16            break;
17        else
18            nghiem = z; %gan x_{n-1} = x_n cho vong lap tiep theo
19            solanlap = solanlap + 1;
20        end
21    end
22 %-----
23    figure('Visible','off');
24    plot(app.UIAxes,x,y,x,x,nghiem,phuongtrinh(nghiem),'r*');
25    app.EditFieldNghiem.Value = sprintf('%.4f',nghiem);
26    app.EditFieldSoLanLap.Value = sprintf('%d',solanlap);
27 end

```

Listing 6: Hàm thực hiện phương pháp lặp

### Tóm tắt thuật toán

**Bước 1:** Chuyển  $f(x) = 0$  về dạng

$$x = g(x);$$

**Bước 2:** Kiểm tra điều kiện

$$|g'(x)| \leq q < 1$$

**Bước 3:** Chọn xấp xỉ ban đầu  $x_0 \in [a, b]$ .

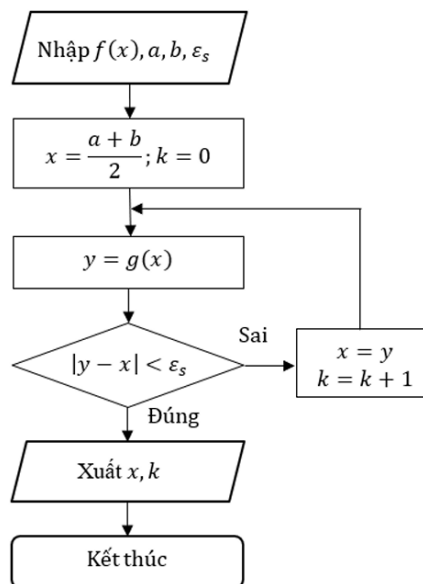
Tính

$$x_n = g(x_{n-1}) \quad n = 1, 2, \dots, k$$

Sau  $k$  lần lặp, được  $x_n \approx \alpha$  là nghiệm gần đúng của phương trình.

Sai số được đánh giá theo công thức (hoặc khi thỏa điều kiện sai số tính toán

$<$  sai số cho phép  $\varepsilon_s$



Hình 3: Thuật toán phương pháp lặp.

```

1 % Button pushed function: ButtonTiepTuyen
2 function ButtonTiepTuyenPushed(app, event)
3     global a;
4     global phuongtrinh;
5     global e;
6     global b;
7 %-----
8     syms V;
9     fd1 = str2func(['@(V)' char(diff(phuongtrinh(V)))]);

```

```

10 x = linspace(a,b,100);
11 y = phuongtrinh(x);
12 nghiem = (a+b)/2;
13 solanlap = 0;
14 while(true)
15     z = phuongtrinh(nghiem)/fd1(nghiem);
16     s = nghiem - z;
17     if abs(s - nghiem) < e
18         break;
19     else
20         nghiem = s;
21         solanlap = solanlap + 1;
22     end
23 end
24 %-----
25 figure('Visible','off');
26 plot(app.UIAxes,x,y,nghiem,phuongtrinh(nghiem),'r*');
27 app.EditFieldNghiem.Value = sprintf('%.4f',nghiem);
28 app.EditFieldSoLanLap.Value = sprintf('%d',solanlap);
29 end

```

Listing 7: Hàm thực hiện phương pháp tiếp tuyến

### Tóm tắt thuật toán

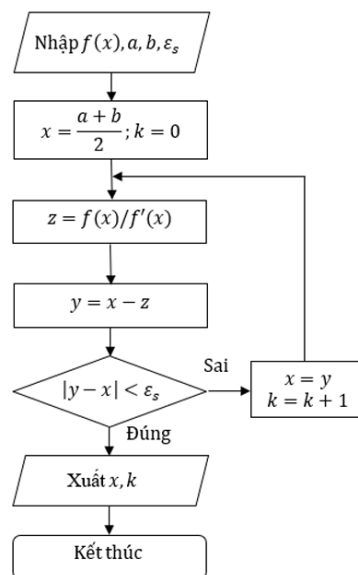
- Bước 1: Tính  $f'(x)$ ,  $f''(x)$  và xét dấu của chúng. Điều kiện để hội tụ  $f'(x)$  và  $f''(x)$  không đổi dấu trên  $[a, b]$
- Bước 2: Chọn  $x_0$  là  $a$  hay  $b$  sao cho  $f(x_0)$  cùng dấu với  $f''(x)$   
 $f(x_0) \cdot f''(x) > 0, \forall x \in [a, b]$ .
- Bước 3: Từ xấp xỉ đầu  $x_0$ , tính

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Sau  $k$  lần lặp ta thu được  $x_k \approx \alpha$  là nghiệm gần đúng của phương trình.

Thường dừng quá trình tính toán khi

$$|x_n - x_{n-1}| < \varepsilon_s \text{ (sai số cho phép)}$$



Hình 4: Thuật toán pháp tiếp tuyến.

```

1 % Button pushed function: ButtonDayCung
2 function ButtonDayCungPushed(app, event)
3     global a;
4     global phuongtrinh;
5     global e;
6     global b;
7 %-----
8     x = linspace(a,b,100);
9     y = phuongtrinh(x);
10    A = a; B = b;
11    solanlap = 0;
12    while (true)
13        nghiem = (A*phuongtrinh(B)-B*phuongtrinh(A))/(phuongtrinh(B)-
14        phuongtrinh(A));
15        if solanlap >= 1

```

```

15         if abs(nghiem - c) < e|x_n - x_{n-1} < saiso|
16             break;
17         end
18     end
19     %thuat toan giống phương pháp chia đôi
20     if phuongtrinh(nghiem)*phuongtrinh(A) < 0
21         B = nghiem;
22     else
23         A = nghiem;
24     end
25     c = nghiem;%gan bien tam c = x_n
26     solanlap = solanlap + 1;
27 end
28 %-----
29 figure('Visible','off');
30 plot(app.UIAxes,x,y,nghiem,phuongtrinh(nghiem),'r*');
31 app.EditFieldNghiem.Value = sprintf('%.4f',nghiem);
32 app.EditFieldSoLanLap.Value = sprintf('%d',solanlap);
33 end

```

Listing 8: Hàm thực hiện phương pháp dây cung

#### Tóm tắt thuật toán

- Bước 1: Tính  $f'(x)$ ,  $f''(x)$  và xét dấu của chúng.  
Hội tụ khi  $f'(x)$  và  $f''(x)$  không đổi dấu trong  $[a,b]$

- Bước 2: Kiểm tra điều kiện

$$\begin{cases} f(x_0).f''(x) < 0 \\ f(d).f(x_0) < 0 \end{cases}$$

Để chọn xấp xỉ ban đầu  $x_0$  là  $a$  hay  $b$

- Bước 3: Từ xấp xỉ ban đầu  $x_0$ , tính

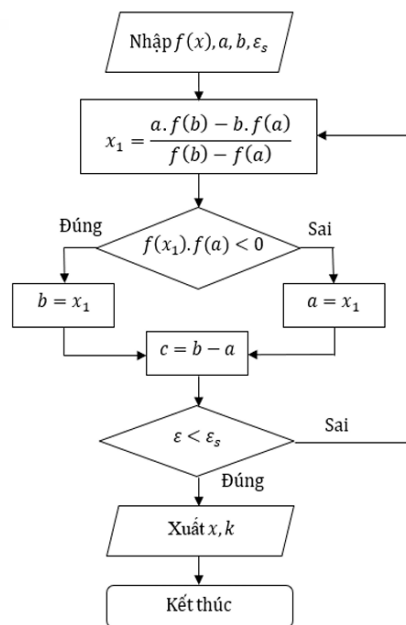
$$x_n = x_{n-1} - \frac{d-x_{n-1}}{f(d)-f(x_{n-1})}f(x_{n-1})$$

Hoặc

$$x_n = \frac{x_{n-1}f(d)-d.f(x_{n-1})}{f(d)-f(x_{n-1})}$$

Sau  $k$  lần lặp ta thu được  $x_k \approx \alpha$  là nghiệm gần đúng của phương trình.

Nếu  $f'(x_0), f''(x)$  cùng dấu: đơn điệu giảm tới nghiệm đúng  $\alpha$  và ngược lại.

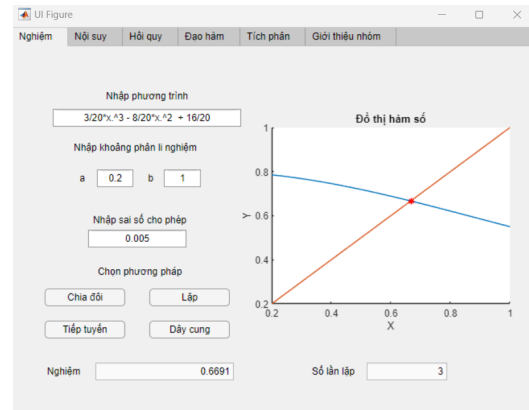
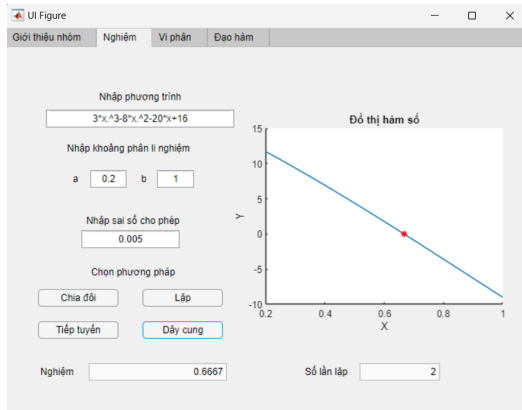


Hình 5: Thuật toán phương pháp dây cung.



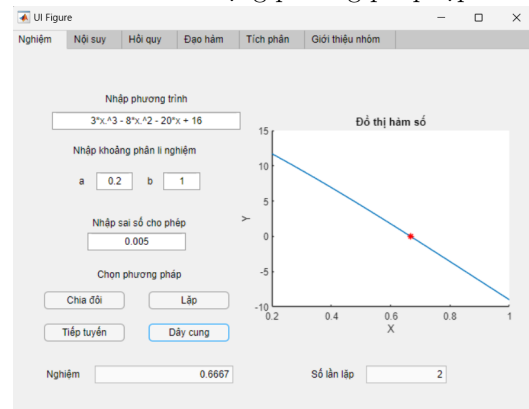
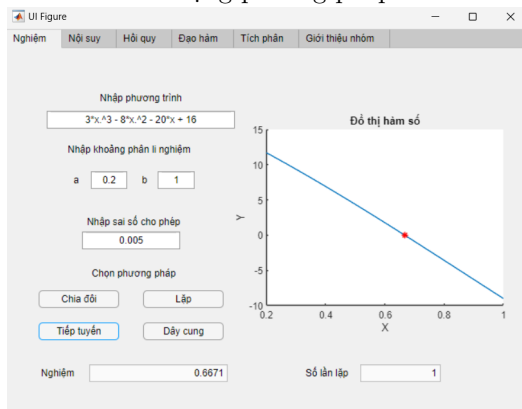
### 1.3 Kiểm tra

Để kiểm chứng, ta sẽ tìm nghiệm của phương trình  $3x^3 - 8x = 20x - 16$  trong khoảng phân li nghiệm  $[0.2, 1]$  với sai số không lớn hơn 0.005.



Hình 6: Sử dụng phương pháp chia đôi

Hình 7: Sử dụng phương pháp lặp



Hình 8: Sử dụng phương pháp tiếp tuyến

Hình 9: Sử dụng phương pháp dây cung

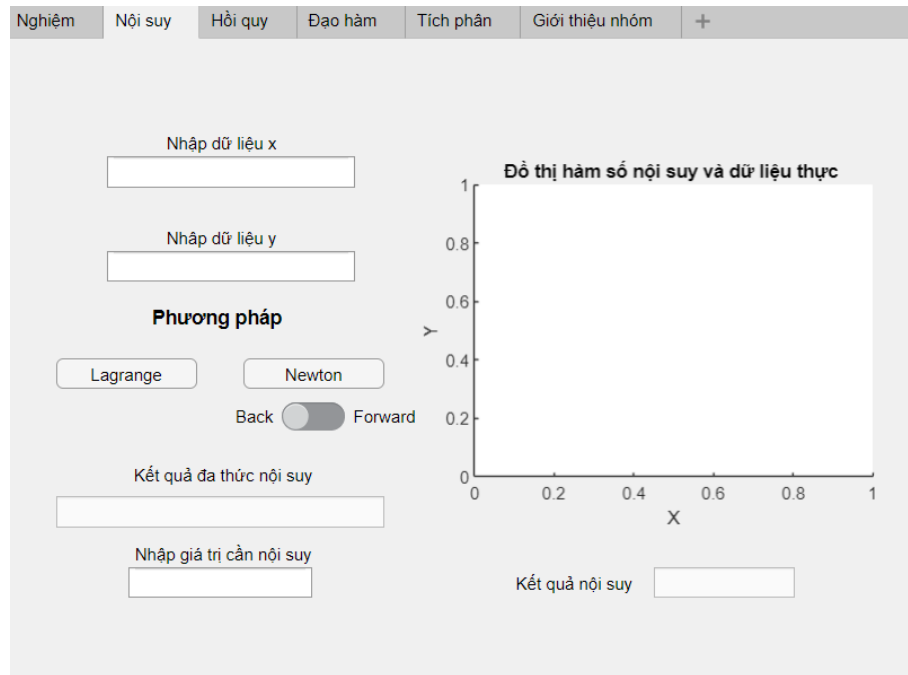
Một điều lưu ý rằng là riêng phương pháp lặp, ta phải nhập phương trình  $g(x)$  ( $x = g(x)$ ):

$$x = \frac{3}{20}x^3 - \frac{8}{20}x^2 + \frac{16}{20}$$

và nghiệm tìm được là được hiển thị là giao điểm của đường thẳng  $y = x$  và  $g(x)$ .

## 2 Nội suy

### 2.1 Giao diện



Hình 10: Giao diện Nội suy

### 2.2 Thuật toán

Để thực hiện các phương pháp nội suy ta cần nhập các giá trị  $x, y$  ở dạng vector. Tuy nhiên ta dùng `EditField` (Text) để nhận giá trị đầu vào với kiểu dữ liệu `char`. Sau đó dùng hàm `strsplit` để tách các số và chuyển sang dạng `double` bằng hàm `str2double`. Các phương pháp Hồi quy, Đạo hàm và Tích phân cũng dùng cách tương tự.

```
1 % Value changed function: EditFieldXa
2 function EditFieldXaValueChanged(app, event)
3     global xa;
4     temp = strsplit(app.EditFieldXa.Value, ' ');
5     xa = zeros(size(temp));
6     for i = 1:length(temp)
7         xa(i) = str2double(temp(i));
8     end
9 end
```

Listing 9: Hàm nhận và xử lý vector  $x$

```
1 % Value changed function: EditFieldYa
2 function EditFieldYaValueChanged(app, event)
3     global ya;
4     temp = strsplit(app.EditFieldYa.Value, ' ');
5     ya = zeros(size(temp));
6     for i = 1:length(temp)
7         ya(i) = str2double(temp(i));
8     end
9 end
```

Listing 10: Hàm nhận và xử lý vector  $y$

```
1 % Value changing function: EditFieldGiaTriNoiSuy
2 function EditFieldGiaTriNoiSuyValueChanged(app, event)
```

```

3   global x0;
4   x0 = app.EditFieldGiaTriNoiSuy.Value;
5   x0 = str2double(x0);
6 end

```

Listing 11: Hàm nhận giá trị nội suy cần tính

```

1 % Value changed function: Switch
2 function SwitchValueChanged(app, event)
3   global ForwardOrBack;
4   ForwardOrBack = app.Switch.Value;
5 end

```

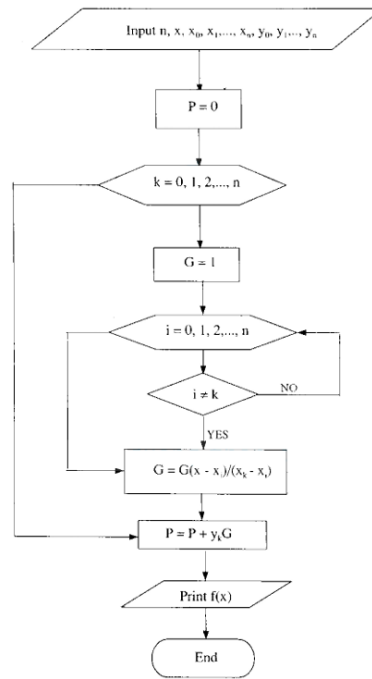
Listing 12: Hàm nhận giá trị ForwardOrBack để tính đa thức nội suy Newton tiến/lùi

```

1 % Value changed function: LagrangeButton
2 function LagrangeButtonValueChanged(app, event)
3   global xa;
4   global ya;
5   global x0;
6 %-----
7   n = length(xa);
8   sum = 0;
9   for i = 1:n
10      product = ya(i);
11      for j = 1:n
12         if i ~= j
13            product = product*(x0 - xa(j))/(xa(i)-xa(j));
14         end
15      end
16      sum = sum + product;
17   end
18   app.EditFieldKetQua.Value = sprintf('%.4f',sum);
19 % %-----
20   syms x;
21   P = 0;
22   for k = 1:n
23      L_k = 1;
24      for i = 1:n
25         if i ~= k
26            L_k = L_k * (x - xa(i)) / (xa(k) - xa(i));
27         end
28      end
29      P = P + ya(k) * L_k;
30   end
31   P = simplify(P); % Rut gon da thuc
32   Pstr = char(P);
33   app.EditFieldDaThucNoiSuy.Value = Pstr;
34 %-----
35   t = linspace(xa(1),xa(end),100);
36   f = matlabFunction(P);
37   figure('Visible','off');
38   plot(app.UIAxes2,t,f(t),x0,f(x0),'r*');
39 end

```

Listing 13: Hàm tính giá trị nội suy và tìm đa thức nội suy Lagrange



Hình 11: Sơ đồ giải thuật phương pháp Lagrange.

```

1 % Button pushed function: NewtonButton
2 function NewtonButtonValueChanged(app, event)
3     global xa;
4     global ya;
5     global x0;
6     global ForwardOrBack;
7 %-----
8     n = length(xa);
9     d = ya;
10    h = xa;
11    if ForwardOrBack == "Back"
12        d = fliplr(d);
13        h = fliplr(h);
14    end
15    for i = 1:n
16        for j = 1:i-1
17            d(i) = (d(j) - d(i)) / (h(j) - h(i));
18        end
19    end
20    n = length(d);
21    result = d(n);
22    for i = n-1:-1:1
23        result = result * (x0 - h(i)) + d(i);
24    end
25    app.EditFieldKetQua.Value = sprintf('%.4f',result);
26 %-----
27    syms x;
28    d = ya;
29    h = xa;
30    if ForwardOrBack == "Back"
31        d = fliplr(d);
32        h = fliplr(h);
33    end
34    P = d(1);
35    % Tính bang hieu chia

```

```

36 divided_differences = zeros(n, n);
37 divided_differences(:,1) = d';
38 for j = 2:n
39     for i = j:n
40         divided_differences(i,j) = (divided_differences(i,j-1) -
divided_differences(i-1,j-1)) / (h(i) - h(i-j+1));
41     end
42 end
43 % Tính đa thức nội suy
44 for i = 1:n-1
45     term = 1;
46     for j = 1:i
47         term = term * (x - h(j));
48     end
49     P = P + divided_differences(i+1, i+1) * term;
50 end
51 P = simplify(P); %Rút gọn đa thức
52 Pstr = char(P);
53 app.EditFieldDaThucNoiSuy.Value = Pstr;
54 %-----
55 t = linspace(xa(1),xa(end),100);
56 f = matlabFunction(P);
57 figure('Visible','off');
58 plot(app.UIAxes2,t,f(t),x0,f(x0),'r*');
59 end

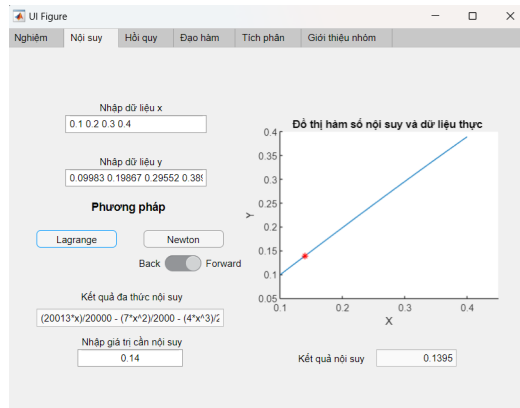
```

Listing 14: Hàm tính giá trị nội suy và tìm đa thức nội suy Newton

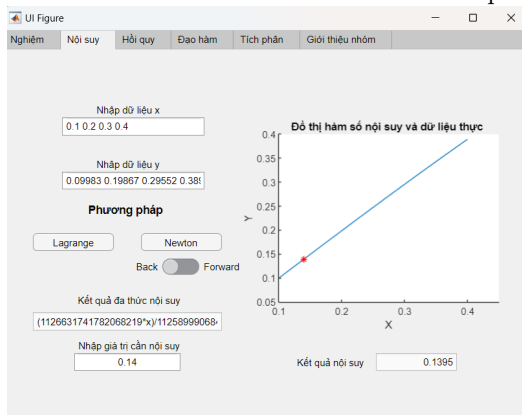
Phương pháp nội suy Newton là phương pháp cải tiến từ phương pháp Lagrange với các bước  $h$  đều và thuật toán được xây dựng dựa trên công thức trong tài liệu. Trong tài liệu đã cho cách tính nội suy Newton tiến, còn nội suy Newton lùi đơn giản là lật ngược vector  $x$  và  $y$  rồi tính như Newton tiến.

## 2.3 Kiểm tra

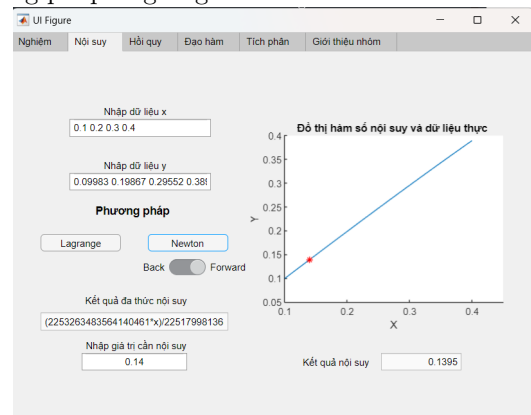
Để kiểm chứng, ta sẽ tìm giá trị nội suy tại  $x = 0.14$  với 2 các giá trị đầu vào cho nội suy:  
 $x = [0.1, 0.2, 0.3, 0.4]$ ,  $y = [0.09983, 0.19867, 0.29552, 0.38942]$ .



Hình 12: Kết quả bằng phương pháp Lagrange



Hình 13: Kết quả bằng phương pháp Newton (lùi)



Hình 14: Kết quả bằng phương pháp Newton (tiến)

- Kết quả đa thức nội suy Lagrange:

$$-\frac{4x^3}{25} - \frac{7x^2}{2000} + \frac{20013x}{20000} - \frac{1}{25000}$$

- Kết quả đa thức nội suy Newton (tiến):

$$-\frac{720575940379333x^3}{4503599627370496} - \frac{1970324836970569x^2}{562949953421312000} + \frac{2253263483564140461x}{2251799813685248000} - \frac{90071992547249}{2251799813685248000}$$

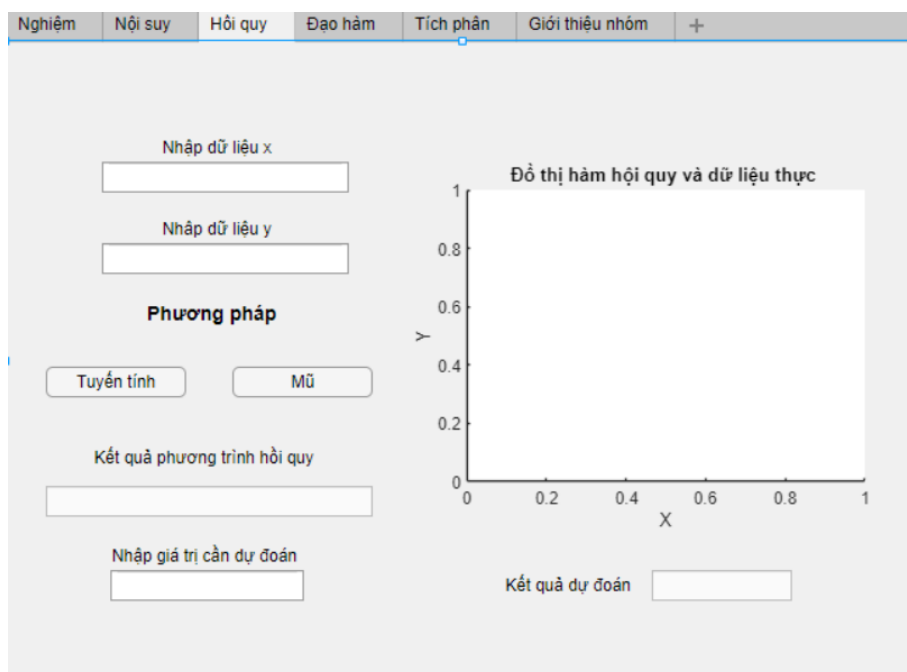
- Kết quả đa thức nội suy Newton (lùi):

$$-\frac{720575940379333}{4503599627370496}x^3 - \frac{788129934787423}{225179981368524800}x^2 + \frac{1126631741782068219}{1125899906842624000}x - \frac{562949953417289}{14073748835532800000}$$

### 3 Hồi quy

Theo như mẫu bao gồm 3 phương pháp là tuyến tính, mũ và logarit nhưng vì phương pháp logarit không được học trong môn Phương pháp tính nên phần này sẽ bị lược bỏ.

#### 3.1 Giao diện



Hình 15: Giao diện Hồi quy

#### 3.2 Thuật toán

```
1 % Value changed function: EditFieldXi
2 function EditFieldXiValueChanged(app, event)
3     global xi;
4     temp = strsplit(app.EditFieldXi.Value, ' ');
5     xi = zeros(size(temp));
6     for i = 1:length(temp)
7         xi(i) = str2double(temp(i));
8     end
9 end
```

Listing 15: Hàm nhận và xử lý vector x

```
1 % Value changed function: EditFieldYi
2 function EditFieldYiValueChanged(app, event)
3     global yi;
4     temp = strsplit(app.EditFieldYi.Value, ' ');
5     yi = zeros(size(temp));
6     for i = 1:length(temp)
7         yi(i) = str2double(temp(i));
8     end
9 end
```

Listing 16: Hàm nhận và xử lý vector y

```

1 % Value changed function: EditFieldGiaTriDuDoan
2 function EditFieldGiaTriDuDoanValueChanged(app, event)
3     global xe;
4     xe = app.EditFieldGiaTriDuDoan.Value;
5     xe = str2double(xe);
6 end

```

Listing 17: Hàm nhận giá trị dự đoán

```

1 % Button pushed function: ButtonTuyenTinh
2 function ButtonTuyenTinhPushed(app, event)
3     global xi;
4     global yi;
5     global xe;
6 %-----
7     z = [sum(xi.^2) sum(xi); sum(xi) length(xi)];
8     y = [sum(xi.*yi); sum(yi)];
9     a = z\y; a = a(1,1);
10    b = z\y; b = b(2,1);
11    syms x;
12    f = @(x) a*x + b;
13 %-----
14    figure('Visible', 'off');
15    t = linspace(xi(1),xi(end),100);
16    app.UIAxes.Visible = true;
17    plot(app.UIAxes3,t,f(t),xi,yi,'r*');
18 %-----
19    % Construct the string manually
20    str_f = sprintf('y = %.4f*x + %.4f', a, b);
21    % Assign the string to the Edit Field
22    app.EditFieldDaThucHoiQuy.Value = str_f;
23 %-----
24    app.EditFieldKetQuaDuDoan.Value = sprintf('%.4f',f(xe));
25 end

```

Listing 18: Hàm thực thi đa thức hồi quy tuyến tính

```

1 % Value changed function: ButtonMu
2 function ButtonMuValueChanged(app, event)
3     global xi;
4     global yi;
5     global xe;
6 %-----
7     z = [sum(xi.^2) sum(xi); sum(xi) length(xi)];
8     y = [sum(xi.*log10(yi)); sum(log10(yi))];
9     A = z\y; A = A(1,1);
10    b = A/log10(exp(1));
11    B = z\y; B = B(2,1);
12    a = 10^B;
13    syms x;
14    f = @(x) a*exp(b*x);
15 %-----
16    figure('Visible', 'off');
17    t = linspace(xi(1),xi(end),100);
18    app.UIAxes.Visible = true;
19    plot(app.UIAxes3,t,f(t),xi,yi,'r*');
20 %-----
21    % Construct the string manually
22    str_f = sprintf('y = %.4f*e^(%.4fx)', a, b);
23    % Assign the string to the Edit Field
24    app.EditFieldDaThucHoiQuy.Value = str_f;

```



```

25 %-----
26 app.EditFieldKetQuaDuDoan.Value = sprintf('% .4f ',f(xe));
27 end

```

Listing 19: Hàm thực thi đa thức hồi quy hàm mũ

### 3.3 Kiểm tra

Thuật toán chủ yếu được xây dựng dựa trên các bước tính dưới đây.

#### 3.3.1 Tuyến tính

VD: Tìm hàm hồi quy xấp xỉ bằng số:

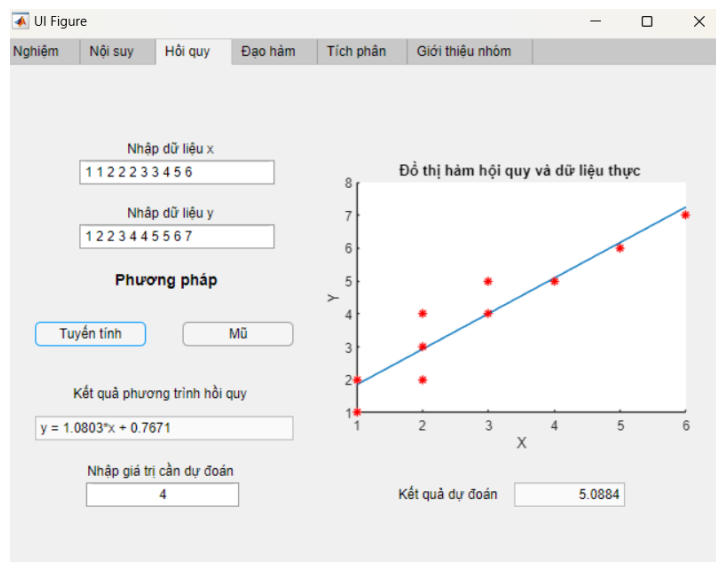
x	1	1	2	2	2	3	3	4	5	6
y	1	2	2	3	4	4	5	5	6	7

Chọn  $y = ax + b$  (đường thẳng)

Giải hệ phương trình:

$$\begin{cases} a \sum x_i^2 + b \sum x_i = \sum x_i y_i \\ a \sum x_i + b.n = \sum y_i \end{cases} = \begin{cases} 109a + 29b = 140 \\ 29a + 10b = 39 \end{cases}$$

$$a = 1.0803, b = 0.7671 \Rightarrow P(x) = 1.0803x + 0.7671$$



Hình 16: Kết quả hồi quy bằng tuyến tính

#### 3.3.2 Hàm mũ

VD: Cho bảng dữ liệu gốc ban đầu:

x	0.65	0.75	0.85	0.95	1.15
y	0.96	1.06	1.17	1.29	1.58

$$P(x) = y = ae^{bx}$$

Log 2 vế ta có

$$\lg(y) = x \cdot \lg e + \lg a$$

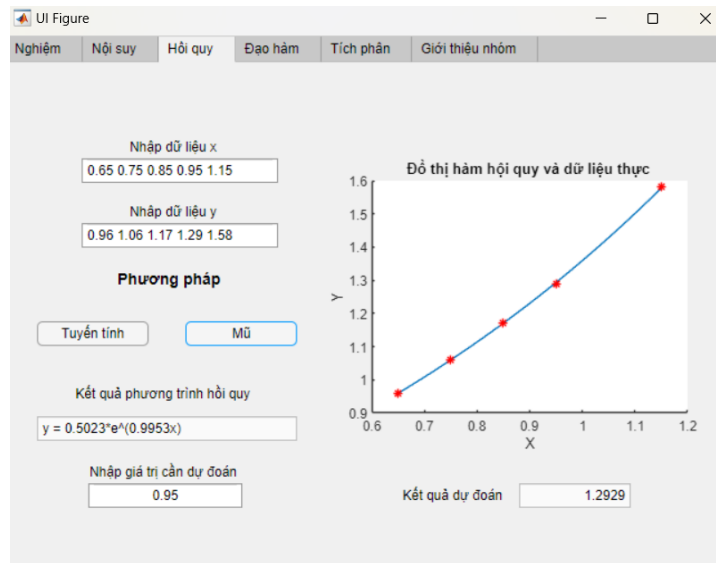
Đặt  $Y = \lg y, A = \lg e, B = \lg a, X = x$

Ta có:  $Y = A.X + B$

$$\begin{cases} A \sum X_i^2 + B \sum X_i = \sum X_i Y_i \\ A \sum X_i + B.n = \sum Y_i \end{cases} \Rightarrow \begin{cases} 3.9325A + 4.35B = 0.3989 \\ 4.35A + 5B = 0.3850 \end{cases} \Rightarrow \begin{cases} A = 0.4321 \\ B = -0.2989 \end{cases}$$

$$A = b \lg e = 0.4321 \Rightarrow b = \frac{0.4321}{\lg e} = 0.995$$

$$B = \lg a \Rightarrow a = 0.5025 \Rightarrow P(x) = 0.5025e^{0.995x}$$



Hình 17: Kết quả hồi quy bằng hàm mũ

## 4 Đạo hàm

### 4.1 Giao diện

Hình 18: Giao diện Đạo hàm

### 4.2 Thuật toán

```

1 % Value changed function: EditFieldGiaTriNoiSuy, EditFieldX
2 function EditFieldXValueChanged(app, event)
3     global X;
4     temp = strsplit(app.EditFieldX.Value, ' ');
5     X = zeros(size(temp));
6     for i = 1:length(temp)
7         X(i) = str2double(temp(i));
    
```

```
8     end
9 end
```

Listing 20: Hàm nhận và xử lý vector x

```
1 % Value changed function: EditFieldY
2 function EditFieldYValueChanged(app, event)
3     global Y;
4     temp = strsplit(app.EditFieldY.Value, ' ');
5     Y = zeros(size(temp));
6     for i = 1:length(temp)
7         Y(i) = str2double(temp(i));
8     end
9 end
```

Listing 21: Hàm nhận và xử lý vector y

```
1 % Value changed function: EditFieldFunc
2 function EditFieldFuncValueChanged(app, event)
3     global f;
4     f = app.EditFieldFunc.Value;
5     f = str2func(['@(x)',f]);
6 end
```

Listing 22: Hàm nhận phương trình

```
1 % Value changed function: SwitchGiatrisaiso
2 function SwitchGiatrisaisoValueChanged(app, event)
3     global Giatrisaiso;
4     Giatrisaiso = app.SwitchGiatrisaiso.Value;
5 end
```

Listing 23: Hàm chọn kiểu giá trị sai số

```
1 % Value changed function: EditFieldGiaTriDaoHam
2 function EditFieldGiaTriDaoHamValueChanged(app, event)
3     global Giatridaoham;
4     Giatridaoham = app.EditFieldGiaTriDaoHam.Value;
5     Giatridaoham = str2double(Giatridaoham);
6 end
```

Listing 24: Hàm nhận giá trị đạo hàm cần tính

```
1 % Button pushed function: ButtonBack
2 function ButtonBackPushed(app, event)
3     global X;
4     global Y;
5     global f;
6     global H;
7     global Giatrisaiso;
8     global Giatridaoham;
9 %-----
10    if ~isempty(app.EditFieldX.Value) && ~isempty(app.EditFieldY.Value)
11        % Neu nhap x va y
12        index = find(X == Giatridaoham);
13        if isempty(index)
14            app.EditFieldKetQuaDaoHam.Value = 'Error!';
15        end
16        h = X(index) - X(index-1);
17        if (Giatrisaiso == "O(h) is chosen")
18            if X(index) >= X(2)
19                approx_derivative = (Y(index) - Y(index-1)) / h;
```

```

20         app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
21     else
22         app.EditFieldKetQuaDaoHam.Value = 'Error!';
23     end
24     elseif (Giatrisaiso == "0(h^2) is chosen")
25         if X(index) >= X(3)
26             approx_derivative = (3*Y(index) - 4*Y(index-1) + Y(index-2))
/ (2*h);
27             app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
28         else
29             app.EditFieldKetQuaDaoHam.Value = 'Error!';
30         end
31     end
32 %-----
33     elseif ~isempty(app.EditFieldFunc.Value)
34         % Neu co nhap ham so
35         if (Giatrisaiso == "0(h) is chosen")
36             approx_derivative = (f(Giatridaoham) - f(Giatridaoham-H)) / H;
37         elseif (Giatrisaiso == "0(h^2) is chosen")
38             approx_derivative = (3*f(Giatridaoham) - 4*f(Giatridaoham-H) + f(
Giatridaoham-2*H)) / (2*H);
39         end
40         app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
41     end
42 %-----
43 end

```

Listing 25: Hàm tính đạo hàm xấp xỉ lùi

```

1 % Button pushed function: ButtonMid
2 function ButtonMidPushed(app, event)
3     global X;
4     global Y;
5     global f;
6     global H;
7     global Giatrisaiso;
8     global Giatridaoham;
9 %-----
10    if ~isempty(app.EditFieldX.Value) && ~isempty(app.EditFieldY.Value)
11        % Neu nhap x va y
12        index = find(X == Giatridaoham);
13        if isempty(index)
14            app.EditFieldKetQuaDaoHam.Value = 'Error!';
15        end
16        h = X(index) - X(index-1);
17        if (Giatrisaiso == "0(h) is chosen")
18            if X(index) > X(1) && X(index) < X(end)
19                approx_derivative = (Y(index+1) - Y(index-1)) / (2*h);
20                app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
21            else
22                app.EditFieldKetQuaDaoHam.Value = 'Error!';
23            end
24        elseif (Giatrisaiso == "0(h^2) is chosen")
25            if X(index) > X(2) && X(index) < X(end-1)
26                approx_derivative = (-Y(index+2) + 8*Y(index+1) - 8*Y(index
-1) + Y(index-2)) / (12*h);

```

```

27         app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
28     else
29         app.EditFieldKetQuaDaoHam.Value = 'Error!';
30     end
31 end
32 %-----
33 elseif ~isempty(app.EditFieldFunc.Value)
34 % Neu co nhap ham so
35 if (Giatrisaiso == "0(h) is chosen")
36     approx_derivative = (f(Giatridaoham+H) - f(Giatridaoham-H)) / (2*
H);
37     app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
38 elseif (Giatrisaiso == "0(h^2) is chosen")
39     approx_derivative = (-f(Giatridaoham+2*H) + 8*f(Giatridaoham+H) -
8*f(Giatridaoham-H) + f(Giatridaoham-2*H)) / (12*H);
40     app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
41 end
42 end
43 end

```

Listing 26: Hàm tính đạo hàm xấp xỉ trung tâm

```

1 % Button pushed function: EditFieldForward
2 function EditFieldForwardButtonPushed(app, event)
3     global X;
4     global Y;
5     global f;
6     global H;
7     global Giatrisaiso;
8     global Giatridaoham;
9 %-----
10 if ~isempty(app.EditFieldX.Value) && ~isempty(app.EditFieldY.Value)
11 % Neu nhap x va y
12 index = find(X == Giatridaoham);
13 if isempty(index)
14     app.EditFieldKetQuaDaoHam.Value = 'Error!';
15 end
16 h = X(index+1) - X(index);
17 if (Giatrisaiso == "0(h) is chosen")
18     if X(index) <= X(end - 1)
19         approx_derivative = (Y(index+1) - Y(index)) / h;
20         app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
21     else
22         app.EditFieldKetQuaDaoHam.Value = 'Error!';
23     end
24 elseif (Giatrisaiso == "0(h^2) is chosen")
25     if X(index) <= X(end - 2)
26         approx_derivative = (- Y(index+2) + 4*Y(index+1) - 3*Y(index)
) / (2*h);
27         app.EditFieldKetQuaDaoHam.Value = sprintf('%.4f',
approx_derivative);
28     else
29         app.EditFieldKetQuaDaoHam.Value = 'Error!';
30     end
31 end
32 %-----

```

```

33 elseif ~isempty(app.EditFieldFunc.Value)
34 % Neu co nhap ham so
35 if (Giatrisaiso == "0(h) is chosen")
36     approx_derivative = (f(Giatridaoham+H) - f(Giatridaoham)) / H;
37 elseif (Giatrisaiso == "0(h^2) is chosen")
38     approx_derivative = (- f(Giatridaoham+2*H) + 4*f(Giatridaoham+H)
- 3*f(Giatridaoham)) / (2*H);
39 end
40 app.EditFieldKetQuaDaoHam.Value = sprintf('% .4f', approx_derivative);
41 end
42 end

```

Listing 27: Hàm tính đạo hàm xấp xỉ tiến

- Trong trường hợp nhập x và y:
  - Bước 1: Kiểm tra giá trị cần tính đạo hàm có trong x.
  - Bước 2: Thực hiện tính toán dựa trên công thức và bắt lỗi. Ví dụ ta cần tính giá trị đạo hàm tại  $x_0$  bằng xấp xỉ lùi thì kết quả sẽ hiển thị là **Error!**.

\* Xấp xỉ lùi:

$$f'(x_i) = \frac{y_i - y_{i-1}}{h} + O(h)$$

$$f'(x_i) = \frac{3y_i - 4y_{i-1} + y_{i-2}}{2h} + O(h^2)$$

\* Xấp xỉ trung tâm:

$$f'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2)$$

$$f'(x_i) = \frac{-y_{i+2} + 8y_{i+1} - 8y_{i-1} + y_{i-2}}{12h} + O(h^4)$$

\* Xấp xỉ tiến:

$$f'(x_i) = \frac{y_{i+1} - y_i}{h} + O(h)$$

$$f'(x_i) = \frac{-y_{i+2} + 4y_{i+1} - 3y_i}{2h} + O(h^2)$$

- Trong trường hợp nhập hàm số và bước nhảy: bỏ qua việc bắt lỗi và chỉ tính dựa trên công thức với  $y = f(x)$ .

### 4.3 Kiểm tra

VD: Cho hàm số :  $y(x) = \arcsin(x)$  với các giá trị tại:  
 $x = [0.1, 0.3, 0.5, 0.7, 0.9]$ ,  $y = [0.1002, 0.3047, 0.5236, 0.7754, 1.1198]$

- Nhập thông số x và y:

UI Figure application interface showing the 'Xấp xỉ lùi' (Backward Approximation) method. The input fields for x and y are filled with the values from the example. The result field shows 1.0945.

Hình 19: Xấp xỉ lùi với sai số  $O(h)$

UI Figure application interface showing the 'Xấp xỉ trung tâm' (Central Approximation) method. The input fields for x and y are filled with the values from the example. The result field shows 1.1305.

Hình 20: Xấp xỉ trung tâm với sai số  $O(h^2)$

UI Figure application interface showing the 'Xấp xỉ trung tâm' (Central Approximation) method. The input fields for x and y are filled with the values from the example. The result field shows 1.1767.

Hình 21: Xấp xỉ trung tâm với sai số  $O(h)$

UI Figure application interface showing the 'Xấp xỉ trung tâm' (Central Approximation) method. The input fields for x and y are filled with the values from the example. The result field shows 1.1442.

Hình 22: Xấp xỉ trung tâm với sai số  $O(h^2)$

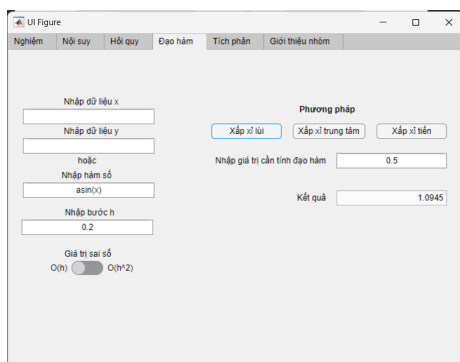
UI Figure application interface showing the 'Xấp xỉ tiến' (Forward Approximation) method. The input fields for x and y are filled with the values from the example. The result field shows 1.2590.

Hình 23: Xấp xỉ tiến với sai số  $O(h)$

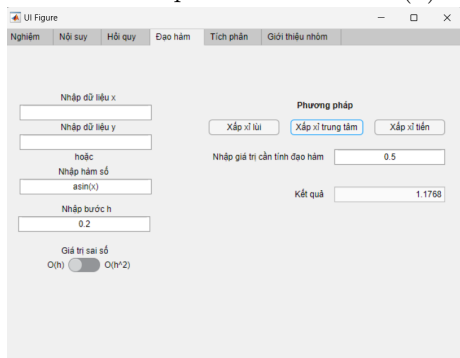
UI Figure application interface showing the 'Xấp xỉ tiến' (Forward Approximation) method. The input fields for x and y are filled with the values from the example. The result field shows 1.0275.

Hình 24: Xấp xỉ tiến với sai số  $O(h^2)$

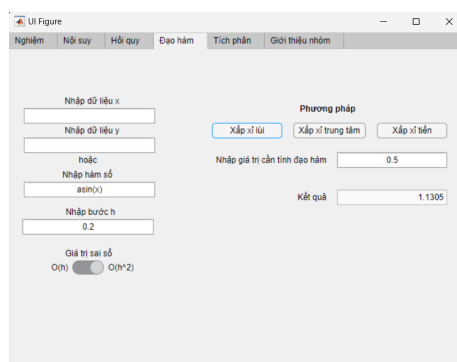
- Nhập phương trình và bước h:



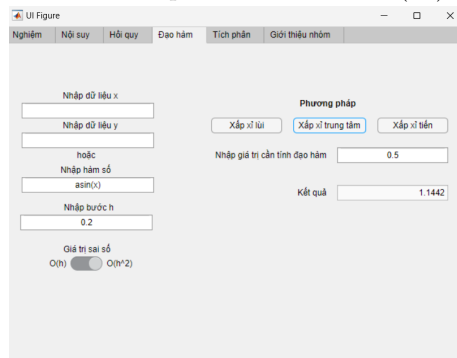
Hình 25: Xấp xỉ lùi với sai số  $O(h)$



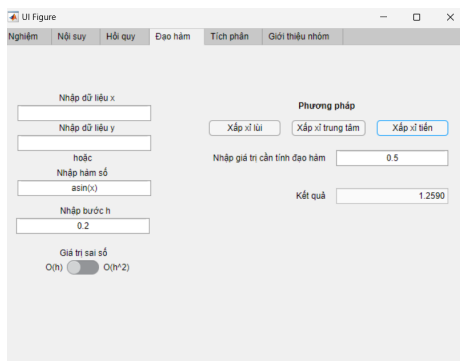
Hình 27: Xấp xỉ trung tâm với sai số  $O(h)$



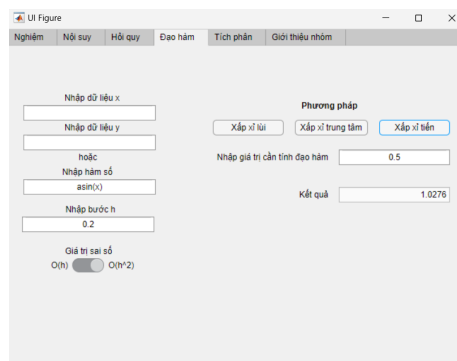
Hình 26: Xấp xỉ lùi với sai số  $O(h^2)$



Hình 28: Xấp xỉ trung tâm với sai số  $O(h^2)$



Hình 29: Xấp xỉ tiến với sai số  $O(h)$



Hình 30: Xấp xỉ tiến với sai số  $O(h^2)$



## 5 Tích phân

### 5.1 Giao diện

Hình 31: Giao diện Tích phân

### 5.2 Thuật toán

```

1 % Value changed function: EditFieldXT
2 function EditFieldXTValueChanged(app, event)
3     global XT;
4     temp = strsplit(app.EditFieldXT.Value, ' ');
5     XT = zeros(size(temp));
6     for i = 1:length(temp)
7         XT(i) = str2double(temp(i));
8     end
9 end

```

Listing 28: Hàm nhận và xử lý vector x

```

1 % Value changed function: EditFieldYT
2 function EditFieldYTValueChanged(app, event)
3     global YT;
4     temp = strsplit(app.EditFieldYT.Value, ' ');
5     YT = zeros(size(temp));
6     for i = 1:length(temp)
7         YT(i) = str2double(temp(i));
8     end
9 end

```

Listing 29: Hàm nhận và xử lý vector y

```

1 % Value changed function: EditFieldFuncT
2 function EditFieldFuncTValueChanged(app, event)
3     global FT;
4     FT = app.EditFieldFuncT.Value;
5     FT = str2func(['@(x)', FT]);

```

6 end

Listing 30: Hàm nhận phương trình

```
1 % Value changed function: EditFieldH
2 function EditFieldHValueChanged(app, event)
3     global H;
4     H = app.EditFieldH.Value;
5     H = str2double(H);
6 end
```

Listing 31: Hàm nhận giá trị bước h

```
1 % Value changed function: EditFieldNhapCan
2 function EditFieldNhapCanValueChanged(app, event)
3     global CanT;
4     temp = strsplit(app.EditFieldNhapCan.Value, ' ');
5     CanT = zeros(size(temp));
6     for i = 1:length(temp)
7         CanT(i) = str2double(temp(i));
8     end
9 end
10
```

Listing 32: Hàm nhận giá trị cận tính tích phân

```
1 % Value changed function: EditFieldNhapN
2 function EditFieldNhapNValueChanged(app, event)
3     global NT;
4     NT = app.EditFieldNhapN.Value;
5     NT = str2double(NT);
6 end
```

Listing 33: Hàm nhận giá trị n

```
1 % Button pushed function: ButtonHinhThang
2 function ButtonHinhThangPushed(app, event)
3     global XT;
4     global YT;
5     global FT;
6     global CanT;
7     global NT;
8 %-----
9     if ~isempty(app.EditFieldXT.Value) && ~isempty(app.EditFieldYT)
10         h = (XT(end) - XT(1)) / (length(XT) - 1);
11         I = h/2 * (YT(1) + YT(end) + 2*sum(YT(2:end-1)));
12 %-----
13     elseif ~isempty(app.EditFieldFuncT.Value)
14         h = (CanT(2) - CanT(1)) / NT;
15         x = CanT(1):h:CanT(end);
16         y = FT(x);
17         I = h/2 * (y(1) + y(end) + 2*sum(y(2:end-1)));
18     end
19 %-----
20     app.EditFieldKetQuaTichPhan.Value = sprintf('%0.4f', I);
21 end
```

Listing 34: Hàm tính tích phân theo phương pháp hình thang

### Công thức hình thang dạng tổng quát

$$I = \int_a^b f(x)dx = \int_{x_0}^{x_1} P_1(x)dx + \int_{x_1}^{x_2} P_1(x)dx + \cdots \int_{x_i}^{x_{i+1}} P_1(x)dx + \cdots + \int_{x_{n-1}}^{x_n} P_1(x)dx$$

$$I = \sum_{i=0}^{n-1} \frac{h}{2} (y_i + y_{i+1}) = \frac{b-a}{2n} (y_0 + y_1 + y_1 + y_2 + \cdots + y_{n-2} + y_{n-1} + y_{n-1} + y_n)$$

$$I = \frac{b-a}{2n} (y_0 + 2y_1 + 2y_2 + \cdots + 2y_{n-1} + y_n)$$

Hình 32: Công thức tích phân hình thang dạng tổng quát

```

1 % Button pushed function: ButtonSimpson13
2 function ButtonSimpson13Pushed(app, event)
3     global XT;
4     global YT;
5     global FT;
6     global CanT;
7     global NT;
8     %-----
9     if ~isempty(app.EditFieldXT.Value) && ~isempty(app.EditFieldYT)
10         h = (XT(end) - XT(1)) / (length(XT) - 1);
11         I = h/3 * (YT(1) + YT(end) + 4*sum(YT(2:2:end-1)) + 2*sum(YT(3:2:end-1)));
12     %-----
13     elseif ~isempty(app.EditFieldFuncT.Value)
14         h = (CanT(end) - CanT(1)) / NT;
15         x = CanT(1):h:CanT(end);
16         y = FT(x);
17         I = h/3 * (y(1) + y(end) + 4*sum(y(2:2:end-1)) + 2*sum(y(3:2:end-1)));
18     end
19     %-----
20     app.EditFieldKetQuaTichPhan.Value = sprintf('%0.4f', I);
21 end

```

Listing 35: Hàm tính tích phân theo phương pháp Simpson's 1/3

Khai triển Taylor (Newton tiến) của hàm bậc 2  $P_2(x)$  ta có tích phân địa phương theo công thức Simpson's 1/3 theo biến  $t$  như sau

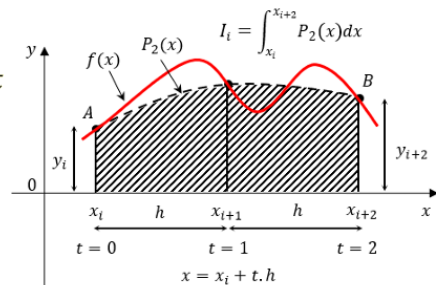
$$I_i = \int_{x_i}^{x_{i+2}} P_2(x)dx \approx \int_0^2 \left[ y_i + t\Delta y_i + \frac{t(t-1)}{2} \Delta^2 y_i \right] h \cdot dt$$

$$I_i = h \left[ y_i t + \frac{t^2}{2} \Delta y_i + \frac{1}{2} \left( \frac{3}{2} - \frac{t^2}{2} \right) \Delta^2 y_i \right] \Bigg|_{t=1}^{t=2}$$

$$I_i = \frac{h}{3} (y_i + 4y_{i+1} + y_{i+2})$$

Ta có công thức Simpson's 1/3 dạng tổng quát ( $i = 0 \div 2m$ ):

$$I = \frac{h}{3} [(y_0 + y_{2m}) + 4(y_1 + y_3 + \cdots + y_{2m-1}) + 2(y_2 + y_4 + \cdots + y_{2m-2})]$$



Hình 33: Công thức tích phân Simpson 1/3.

```

1 % Button pushed function: ButtonSimpson38

```

```

2 function ButtonSimpson38Pushed(app, event)
3     global XT;
4     global YT;
5     global FT;
6     global CanT;
7     global NT;
8     %-----
9     if ~isempty(app.EditFieldXT.Value) && ~isempty(app.EditFieldYT)
10         if mod(length(XT)-1, 3) ~= 0
11             app.EditFieldKetQuaTichPhan.Value = "Error!";
12         else
13             if (length(XT)-1) / 3 == 1
14                 h = (XT(end) - XT(1)) / (length(XT) - 1);
15                 I = (3*h/8) * (YT(1) + 3*YT(2) + 3*YT(3) + YT(end));
16             else
17                 h = (XT(end) - XT(1)) / (length(XT) - 1);
18                 I = (3*h/8) * (YT(1) + 3*sum(YT(2:3:end-1)) + 3*sum(YT(3:3:
19 end-1)) + 2*sum(YT(4:3:end-1)) + YT(end));
20                 app.EditFieldKetQuaTichPhan.Value = sprintf('%.4f', I);
21             end
22         %-----
23     elseif ~isempty(app.EditFieldFuncT.Value)
24         if mod(NT, 3) ~= 0
25             app.EditFieldKetQuaTichPhan.Value = "Error!";
26         else
27             h = (CanT(end) - CanT(1)) / NT;
28             x = CanT(1):h:CanT(end);
29             y = FT(x);
30             I = (3*h/8) * (y(1) + 3*sum(y(2:3:end-1)) + 3*sum(y(3:3:end-1)) +
31 2*sum(y(4:3:end-1)) + y(end));
32             app.EditFieldKetQuaTichPhan.Value = sprintf('%.4f', I);
33         end
34     end
35 end

```

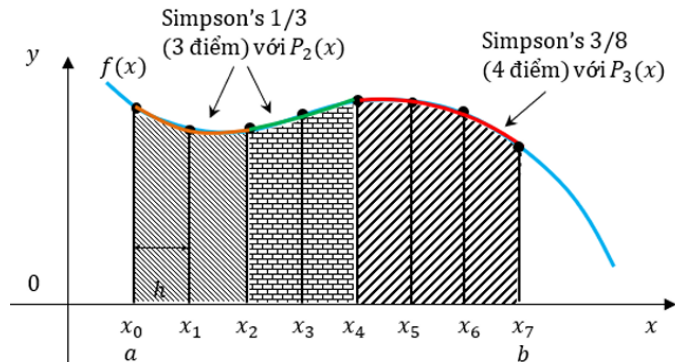
Listing 36: Hàm tính tích phân theo phương pháp Simpson's 3/8

Tương tự như công thức Simpson's 1/3, ta thay hàm số  $f(x)$  bằng đa thức  $P_3(x)$  bậc 3 đi qua 4 điểm. Dùng đa thức nội suy Lagrange bậc 3 ta có dạng tích phân Simpson's 3/8:  $I = \int_a^b f(x)dx \approx \int_a^b P_3(x)dx$

$$I \approx \frac{3h}{8} [y_0 + 3y_1 + 3y_2 + y_3]$$

Trong trường hợp  $h = \frac{b-a}{3}$

$$I \approx \frac{(b-a)}{8} [y_0 + 3y_1 + 3y_2 + y_3]$$



Hình 34: Công thức tích phân Simpson 3/8.

## 5.3 Kiểm tra

### 5.3.1 Phương pháp hình thang

VD: Cho hàm  $y = x^3 \sin(x) + x \cos(x)$  với các thông số:

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	0	0.0996	0.1976	0.2946	0.3933	0.4987	0.6172	0.7564	0.9247	1.1305	1.3818

The screenshot shows the 'UI Figure' application window. The 'Phương pháp' (Method) dropdown is set to 'Hình thang' (Trapezoidal). The 'Nhập dữ liệu x' (Enter x data) field contains the values '0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0'. The 'Nhập dữ liệu y' (Enter y data) field contains the values '0 0.0996 0.1976 0.2946 0.3933 0'. The 'Nhập hàm số' (Enter function) field is empty. The 'Nhập cận để tính tích phân' (Enter limits for integration) field is empty. The 'Nhập N' (Enter N) field is empty. The 'Kết quả' (Result) field shows the value '0.5604'.

Hình 35: Kết quả của nhập x và y

The screenshot shows the 'UI Figure' application window. The 'Phương pháp' (Method) dropdown is set to 'Hình thang' (Trapezoidal). The 'Nhập dữ liệu x' (Enter x data) field is empty. The 'Nhập dữ liệu y' (Enter y data) field is empty. The 'Nhập hàm số' (Enter function) field contains the expression 'x^3.\*sin(x)+x.\*cos(x)'. The 'Nhập cận để tính tích phân' (Enter limits for integration) field contains the value '0 1'. The 'Nhập N' (Enter N) field contains the value '10'. The 'Kết quả' (Result) field shows the value '0.5603'.

Hình 36: Kết quả của nhập hàm  $y = x^3 \sin(x) + x \cos(x)$

### 5.3.2 Phương pháp Simpson's 1/3

VD: Tương tự ví dụ phương pháp hình thang.

The screenshot shows the 'UI Figure' application window. The 'Phương pháp' (Method) dropdown is set to 'Simpson 1/3'. The 'Nhập dữ liệu x' (Enter x data) field contains the values '0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0'. The 'Nhập dữ liệu y' (Enter y data) field contains the values '0 0.0996 0.1976 0.2946 0.3933 0'. The 'Nhập hàm số' (Enter function) field is empty. The 'Nhập cận để tính tích phân' (Enter limits for integration) field is empty. The 'Nhập N' (Enter N) field is empty. The 'Kết quả' (Result) field shows the value '0.5589'.

Hình 37: Kết quả của nhập x và y

The screenshot shows the 'UI Figure' application window. The 'Phương pháp' (Method) dropdown is set to 'Simpson 1/3'. The 'Nhập dữ liệu x' (Enter x data) field is empty. The 'Nhập dữ liệu y' (Enter y data) field is empty. The 'Nhập hàm số' (Enter function) field contains the expression 'x^3.\*sin(x)+x.\*cos(x)'. The 'Nhập cận để tính tích phân' (Enter limits for integration) field contains the value '0 1'. The 'Nhập N' (Enter N) field contains the value '10'. The 'Kết quả' (Result) field shows the value '0.5589'.

Hình 38: Kết quả của nhập hàm  $y = x$

### 5.3.3 Phương pháp Simpson's 3/8

VD: Cho hàm  $y = x^3 \sin(x) + x \cos(x)$  với các thông số:

x	0	0.1111	0.2222	0.3333	0.4444	0.5556	0.6667	0.7778	0.8889	1.0000
y	0	0.1106	0.2192	0.3271	0.4390	0.5624	0.7071	0.8843	1.1055	1.3818

Hình 39: Kết quả của nhập x và y

Hình 40: Kết quả của nhập hàm  $y = x^3 \sin(x) + x \cos(x)$